

Fusing Physical Process Models with Measurement Data Using FIR Calibration ^{*}

Enso Ikonen ^{*} István Selek ^{*}

^{*} *Intelligent Machines and Systems, University of Oulu, FIN-90014
Oulun yliopisto, Finland (e-mail: Enso.Ikonen@oulu.fi;
Istvan.Selek@oulu.fi).*

Abstract:

Tuning of physical plant models is considered in the context of monitoring and control of industrial processes. The problem of calibrating physical models is discussed. A method is proposed, consisting of extending the physical model with finite impulse response (FIR) filters at the output. The approach is illustrated in extensive simulations and applied in state estimation using a nonlinear chemical continuous stirred tank reactor benchmark.

Keywords: dynamic models; machine learning; physical models; process control; process models; state estimation

1. INTRODUCTION

Modern design of process control and monitoring relies heavily on dynamic simulation models. On-line simulations can be necessary for evaluating model predictive controllers, model-based state-estimators, or in model-based fault detection and isolation. Optimization of plant short term dynamic operation or integrated plant and control design are other examples of applications requiring a proper dynamic process simulator. A modern digital twin gets its essential behavioral contents from a plant model, linking simulations with on-line data and various services such as data validation, visualization, process analytics, what-if analysis and 3D plant animations, for example. In this paper, the focus is on process monitoring and control.

In the era of IoT (Internet of Things) and ML/AI (machine learning / artificial intelligence), the role of data-driven modeling has been emphasized. A vast literature exists on proposed data-driven approaches to process modeling as well as experiences on their application to simulated and/or real-life plants (Sjoberg et al. (1995); Hastie et al. (2017); Ikonen and Najim (2002)). The approaches include various basis-function constructions proposed suitable for process modelling tasks, statistical analytics on conclusions that can be drawn from data, and algorithms providing optimal active exploration, just to mention few. Data-driven modeling is plagued by the bias-variance dilemma, related to suppression of noise in measurements and extracting a proper representation of the underlying nonlinear mapping. Both problems are impossible to solve using data alone. The impact of noise can be reduced by taking repeated samples, but the uncertainty will remain. If only sampled data is available, there is no information about the mapping between samples, in time nor space. Consequently, interpolation/extrapolation, i.e. generalization, is

always based on some additional assumptions outside of the data.

The application of physical models is widespread in many fields of engineering (Skogestad (2009)). Physical models provide justified and transparent predictions and explanations for the outcomes, based on mass, momentum and energy balances, laws of mechanics, chemistry, etc. The role of physical models is pronounced in the heavy process industry, where models are extensively used in plant design and experimentation and exploration with plants is severely complicated in the industrial practice, due to costs, slowness of processes and safety issues. However, construction of physical models may also require significant resources and some tuning of such models to local site conditions is always necessary.

The essential component of ML/AI is that of learning from examples, such as process measurement data. Therefore, the topics of model tuning and learning are closely entwined. A recent trend in process control literature is to fuse machine learning with physical models (Bikmukhmetov and Jächke (2020); Qin and Chiang (2019)), with a recognized aim to simultaneously improve the automatic adaptation of physical process models to local conditions and the explainability of data-driven models. This is also the topic of this paper. Compared to existing works, our focus is in this paper to maximally exploit the knowledge in physical models/simulators, and to emphasize the importance of system dynamics. Less weight is put on validating new inventive ML/AI-driven methodologies, as simplicity often comes with robustness and ease of implementation.

This paper considers the problem of identifying an I/O model for a MIMO system, such as

$$\begin{aligned} \mathbf{x}(k+1) &= f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) \\ \mathbf{y}(k) &= g(\mathbf{x}(k), \mathbf{v}(k)) \end{aligned} \quad (1)$$

where $f : R^{n_x} \times R^{n_u} \times R^{n_w} \rightarrow R^{n_x}$ is the state propagation function and $g : R^{n_x} \times R^{n_v} \rightarrow R^{n_y}$ the measurement

^{*} The work in this paper was partly funded by the H2020 project COGNITWIN (grant number 870130).

function. For brevity, the sampling time is assumed to be a constant and the important topics of input/output selection and design of sampling are not considered here. The inputs \mathbf{u} are assumed to be known and measured, possible uncertainties in inputs can be included in the random state noise \mathbf{w} . States \mathbf{x} need not be known, measured or estimated. The plant outputs are measured, corrupted by random noise \mathbf{v} . Noise is assumed to have finite variance, smoothness or existence of derivatives of f or g is not assumed. It is assumed that the system is time-invariant, or that it varies with time very slowly so that (some kind of) ergodic assumptions hold.

This paper proposes an approach based on the use of physical modeling and finite impulse response (FIR) filters. The main benefits are in that the plant models (simulators) can be incorporated into the proposed structure and that the parameter estimation in the data-driven components is particularly simple and robust, enabling automatic procedures. The few parameters to be set by the user are very transparent and their tuning does not require an experienced process control engineer. Section 2 considers calibration of physical plant models and leads to propose a PMFIR structure for model calibration in Section 3. Applications in state estimation and control are briefly discussed in Section 4. Section 5 illustrates an approach using the well known van der Vusse continuous stirred tank reactor (CSTR) benchmark. Discussion and conclusions end the paper.

2. CALIBRATION OF PHYSICAL MODELS

Suppose that a given plant has a dynamic I/O model, such as a state space model:

$$\begin{aligned} \mathbf{x}(k+1) &= f_{PM}(\mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{y}_{PM}(k) &= g_{PM}(\mathbf{x}(k)) \end{aligned} \quad (2)$$

$f_{PM} : R^{n_x} \times R^{n_u} \rightarrow R^{n_x}$ and $g_{PM} : R^{n_x} \rightarrow R^{n_y}$. This model, called a “physical model”, PM, is likely to have been derived based on first principles or detailed engineering knowledge. It may be complex, e.g., due to size or amount of detail, and nothing can be assumed about its internal structure or parameters. It is convenient to simulate the propagation of \mathbf{y}_{PM} in real time in parallel with the plant, or using historical data. The plant model is assumed to have been calibrated as found appropriate and feasible given the constraints on knowledge and resources. It is provided “as is”. This kind of models commonly emerge in industrial plant and/or control design.

Suppose also, that corresponding I/O measurements from a real plant are available: $\mathbf{u}(i)$ and $\mathbf{y}(i)$, $i = 1, 2, \dots, k$. This data is to be used to calibrate the process model, so that the predictions would be consistent with the data. The need for tuning may be due to various reasons, such as correction of simplifications/idealizations made during modeling or adaptation to systematic errors in plant measurements. The purpose of adjustments is to match the physical model and measurements from the local site, in this sense the true state of the plant is irrelevant.

This wish to calibrate the physical model is constrained by a number of engineering requirements. i) Very often it is not desired to tune the parameters inside f_{PM} or g_{PM} , in order to preserve the original model. This is

due to various reasons: these parameters may appear in a complex manner, the plant model is trusted/accepted by the users, or the users are familiar with the model with its defects. ii) The calibration must be automatic from data and should scale well. This requires needing minimal insight to the physical model, and robustness vs. measurement noise and structural errors. iii) The algorithm should be simple to implement and execute on-line in a distributed control system (DCS) environment or alike. iv) The approach should support the a priori assumption that industrial process models are often more accurate in their steady-state description, typically in the focus of plant design. v) The purpose of calibration is not to model phenomena/behaviour that is not described by the physical model. A more detailed plant modeling (of any unmodelled behaviour) is almost surely a project requiring an approach different from an automated procedure. However, it is clear that the borderline between modeling and tuning is somewhat vague. Therefore, vi) the procedure should give some means to analyze the adjustments made to the physical model mapping due to calibration.

Joint physical/data-driven structures (see also Bikmukhametov and Jächke (2020)) for modeling dynamic processes can be roughly categorized into the following:

- (1) A data-driven model (DM) is placed in parallel with the physical model (PM). The outcomes of the two models can be compared by running models in like situations, and the multimodel information then presented to the end-users. Typically, data-driven approaches can provide an estimate of the prediction uncertainty. If such uncertainty estimate is available also for the PM (e.g., via Monte Carlo), some sort of bayesian reasoning can be conducted to devise a more likely estimate.
- (2) A DM is trained to minimize the residual between measurements and the PM output. The overall outcome is a superposition of the PM and DM. This approach is appealing in that the DM is used to compensate for the defects in PM predictions, the behaviour already modeled by PM need not be identified by DM.
- (3) The PM is placed in series with the DM. The DM can precede or follow the PM. DM optimization aims at tuning both static and dynamic behaviour of the PM input/output signals, minimizing the residual between the final prediction and output data. In the linear case this leads to the well known Wiener/Hammerstein structures, common in the industry due to their robustness (Lawrynczuk (2016); Ikonen and Najim (2001)).

A basic analysis of some of the differences of the structures is simple to derive in the linear case (Ikonen and Sele (2020)). Suppose that the physical model dynamics are given by $\frac{num_{PM}}{den_{PM}}$ and the plant data is obtained from $\frac{num_D}{den_D}$. Let the DM minimize the difference between DM output and measured plant D output. The first approach leads to estimate $\frac{num_{DM}}{den_{DM}} \approx \frac{num_D}{den_D}$, while the second and third lead to estimate more complicated dynamics: $\frac{num_{DM}}{den_{DM}} \approx \frac{num_D den_{PM} - num_{PM} den_D}{den_D den_{PM}}$ and $\frac{num_{DM}}{den_{DM}} \approx \frac{num_D}{den_D} \frac{den_{PM}}{num_{PM}}$, respectively. However, the first approach

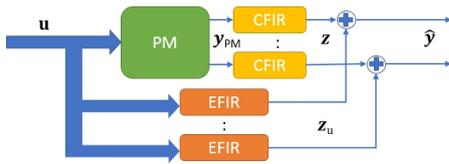


Fig. 1. PMFIR structure for physical model calibration using FIR filters.

provides no remedy for the PM calibration problem, it merely provides a data-driven alternative DM.

An alternative to the above ARX $\frac{num_{DM}}{den_{DM}}$ (autoregressive exogenous) is to estimate only the num_{DM} (finite impulse response, FIR) (Ljung (1997), p. 11; Ikonen and Najim (2002), p. 47):

$$y(k) = b_0 u(k) + b_1 u(k-1) + \dots + b_{n-1} u(k-n-1) \quad (3)$$

The FIR parameters can be estimated using the least squares method (LS), a number of extensions have been proposed to handle the nonparsimonious nature of FIR (Nikolaou and Vuthandam (1998)). The FIR filters are known to be able to model complex dynamics. However, this comes at the price of a large number of parameters to estimate: for FIR the $\dim(num)$ easily becomes much larger than $\dim(num) + \dim(den)$ for ARX. However, the number of parameters in ARX modeling can also be higher than expected at first glance, as was illustrated above. If, in addition, the models are to be used for predicting to the future, the ARX modeling is not feasible, but one needs to turn to output error (OE) modeling. Unfortunately, OE models can not be solved with LS, but require iterative approaches (RPEM, or alike) (Ikonen and Najim (2002), Ch. 3.3).

In conclusion, FIR modeling can be seen as an attractive structure for the problem at hand. This is further confirmed by the fact that a number of model-predictive control approaches use the FIR or finite step response descriptions for the plant (Richalet et al. (1978); Cutler and Ramaker (1979); Garcia and Morshedi (1986)).

Many algorithms have been proposed for data-driven modeling of nonlinear dynamic processes (Sjoberg et al. (1995); Ikonen and Najim (2002); Hastie et al. (2017)). Most -if not all- model structures can be interpreted as basis-function approaches. A physical model provides one possible way to generate basis functions (e.g., in PM followed by DM structure). In fact, since the PM states \mathbf{x} are often accessible as well, they could be used as a more extensive set of basis functions. This is likely to lead to collinearity problems, but they are routinely solved in machine learning (Hastie et al. (2017)). However, this line of thinking is now not pursued further here. A simpler approach is proposed instead, where the PM is expected to provide the essential nonlinear gains/dynamics of the plant via its output predictions. This basic behaviour is then tuned by adjusting the parameters of additional linear FIR filters.

3. PMFIR APPROACH

The above discussion leads to consider the two-phase structure depicted in Fig. 1. The PMFIR model is evaluated in three phases. First, the physical model PM is

simulated using the given input data (either on-line or a series of batch data), providing the PM predictions \mathbf{y}_{PM} . In a second phase the calibration FIR (CFIR) filters, z_i , are evaluated, and the extended FIR (EFIR) filters, $z_{u,i}$, are evaluated in parallel. Finally, the outputs of the two filters are summed, for each output, to provide the calibrated predictions, \hat{y}_i .

The parameters for the CFIR and EFIR need to be determined by the proposed method. The parameter estimation starts by first evaluating the PM predictions \mathbf{y}_{PM} for the training data. Using these as inputs to the CFIR, the optimal CFIR parameters are solved by minimizing the sum of squared residuals between training data and CFIR predictions. The EFIR parameters are then solved by minimizing the sum of squares of the remaining residuals.

Next subsections detail the parameter estimation algorithm for batch data using LS. Algorithms for recursive LS with various forgetting schemes are well known and straightforward to devise.

3.1 Calibration FIR - CFIR

Since the physical model (f_{PM}, g_{PM}) is to be evaluated “as is” using the known inputs $\mathbf{u}(k)$, the physical model is evaluated first. The physical model outputs y_{PM} are calibrated by single-input single-output (SISO) FIR filters at each output of the PM

$$z(k) = b_0 + b_1 y_{PM}(k) + \dots + b_n y_{PM}(k-n+1) \quad (4)$$

where $y_{PM}(k)$ is one of the n_y outputs in the vector \mathbf{y}_{PM} (subscripting is omitted for simplicity). Note that the filter includes a bias term b_0 . This, together with the fact the FIR gain is not constrained, implies that the output can be scaled and shifted in an affine manner.

Each filter has $n+1$ coefficients to estimate. Since FIR parameter estimation is parsimonious (Nikolaou and Vuthandam (1998)), it can be justified to consider regularized approaches when n is considerably large. Represent each SISO FIR model (4) by $z(k) = \theta' \phi(k)$, where the parameter vector θ consists of the unknown coefficients b_0, b_1, \dots, b_n and the regression vector $\phi(k)$ of the corresponding physical model outputs $y_{PM}(k), y_{PM}(k-1), \dots, y_{PM}(k-n)$. Collect the regression vectors into matrix Φ and corresponding measured outputs to a vector \mathbf{Y} . The goal, for each SISO FIR filter, is then to minimize

$$(\mathbf{Y} - \Phi\theta)'(\mathbf{Y} - \Phi\theta) + r\theta' \mathbf{Q}\theta \quad (5)$$

where $r \geq 0$ and \mathbf{Q} is a positive definite matrix. The solution is given by

$$\hat{\theta} = [\Phi_0' \Phi_0 + r\mathbf{Q}]^{-1} \Phi_0' \mathbf{Y}_0 \quad (6)$$

where the subscript 0 in Φ and \mathbf{Y} is used to specify the data set used for parameter estimation (training data). A separate estimation is conducted for each output filter. A solution to the parameter estimation problem exists and is unique provided that the input data is persistently exciting. An increasing weighing for deviations from zero for increasingly delayed elements is obtained e.g. by choosing $\mathbf{Q} = \tilde{N} \text{diag}([1, 2, \dots, n])$ (in the SISO case with no b_0) and r is some small number (e.g., $r = 0.1$). If the physical model static performance is fully trusted (not

to be calibrated), the steady state gain of the FIR can be constrained to 1. A recursive parameter estimation algorithm can be devised, e.g., from the Kalman filter.

3.2 Extended FIR - EFIR

The CFIR-calibrated PM can be further extended with a linear FIR model for the remaining residuals (EFIR). Consider the following multiple-input MISO FIR structure with filter length n for each considered input u_1, u_2, \dots, u_p :

$$z_u(k) = b_{u,0} + b_{u,1}u_1(k-1) + \dots + b_{u,n}u_1(k-n) + \dots + b_{u,np}u_p(k-n). \quad (7)$$

Note that the model causality is preserved by the one sample delay in the inputs. The batch and recursive parameter estimation can be conducted as discussed above, with the exception that the targeted output is the residual error $y(k) - z(k)$, instead of the measured output $y(k)$ (multiple-output indexes omitted for brevity).

3.3 Physical model FIR - PMFIR

The final calibrated PM outputs are then given by

$$\hat{y}_i(k) = z_i(k) + z_{u,i}(k) \quad (8)$$

$i = 1, 2, \dots, m$, where m is the number of calibrated outputs.

The length of FIR filters n is to be chosen by the modeller. This parameter can be set based on knowledge of plant settling time (e.g., from plant step responses), or optimized, e.g. by minimizing the Akaike Information Criterion (AIC) $-2\ln(L) + 2n'$, where L is the maximum value for the likelihood of the model and n' is the number of parameters to estimate. In a variant (Gatti (2005), p. 301; Das (2016)) applicable for least squares regression with normally distributed residuals, $\sum_{k=1}^K (y(k) - \hat{y}(k))^2 + 2n' + \frac{2n'(n'+1)}{K-n'-1}$, the last term is a modification for small sample sizes.

The two-phase estimation emphasizes the PM-calibration aspect of the approach. All nonlinearities in PMFIR originate from the physical model PM. The CFIR provides a linear dynamic adjustment for the PM outputs. The EFIR extends the tuning by a linear dynamic mapping from past PM inputs, thereby enabling identification of linear components not modelled by the PM. If necessary for increased numerical precision, it would seem natural to devise further versions by replacing the EFIR (or CFIR) by a nonlinear time-series estimator. However, as discussed in previous Sections, the interpretability of the adjustments in such approaches may easily suffer.

4. PROCESS STATE ESTIMATION AND CONTROL

Model-based state estimation and control is regularly used in the advanced industrial practice. While it is not reasonable to discuss process control in detail in the context of this paper, it suffices to note that state estimation is an essential part of state feedback control, whether linear or nonlinear. Of course, monitoring applications are justified by their own right in the industrial practice, and can make great use of state estimation techniques.

Bayesian state estimation is a commonly used tool for estimating system states, for fusion of dynamic plant models and noisy measurements. The famous Kalman filter (Simon (2006)) assumes a linear plant description. The extended Kalman filter (EKF) (Simon (2006)) is based on on-line linearization via computation of Jacobians. The outcomes are optimal in an environment corrupted by Gaussian noise, but the Kalman filter provides the optimal linear filter also in a non-Gaussian context (see Simon (2006), p. 130). Recent advances in state estimation can make a more direct use of nonlinear physical models. Despite of the advantages of particle filter (PF) (Gordon et al. (1993); Simon (2006), Ch. 15), such as the ability to deal with complex multimodal densities, the computational load associated with PF can make it infeasible in practice. The unscented Kalman filter (UKF) (Julier et al. (1995); Simon (2006), Ch. 14) provides an approximation of the mean and covariance for nonlinear systems which is often sufficient and reasonable for practical purposes. The estimation can also be constrained so as to ensure feasible estimates at all times (Spivey (2010)).

The UKF is particularly useful in the PMFIR setup where the PM is given in a state space form, eq. (2), as estimation of unknown states in PM is feasible. As one of the options, the calibrated model provides a means to back-calculate the equivalent measurements at the PM level. For the measurement corresponding to y_{PM} it is straightforward to derive

$$q_{PM}(k) = \frac{1}{b_1} [y(k) - z_u(k) - z(k)] + y_{PM}(k) \quad (9)$$

for the SISO case (indexes omitted for brevity). Note that this simple method requires that $b_1 \neq 0$, which may not always be the case. This approach aims at removing the impact of the calibration of the the PMFIR innovation signal. A Kalman filter (extended, unscented, or particle filter) type of estimator can be constructed on the basis of predictions by the process model \mathbf{y}_{PM} and the back-calculated measurement \mathbf{q}_{PM} . An estimate of the states of PM will then be adjusted with the available measurement, consistent with both the calibrated simulation and the physical model internal construction.

The UKF algorithm is well known, and it is not repeated here (see e.g. Simon (2006), Ch. 14.3). It is worthwhile to point out, however, that the one-step ahead simulations from given initial states (determined by the UKF sigma points) are straightforward to propagate and measure, given access to (f_{PM}, g_{PM}) . The number of model evaluations required by UKF is relative to the number of states n_x (not exponential) and the $2n_x$ one-step ahead simulations can be expected to remain feasible in many cases of industrial size.

5. NUMERICAL CSTR BENCHMARK EXAMPLE

The PMFIR approach was tested using the well known process control design benchmark of a highly nonlinear continuous stirred tank reactor (CSTR) with a cooling jacket (Engel and Klatt (1993); Chen et al. (1995); Kravaris et al. (1998); Gatzke and Doyle (1999); Perez et al. (2002); Garcia-Gabin et al. (2006); Ikonen et al. (2016); Marusak (2020)). The system, see Fig. 2, exhibits

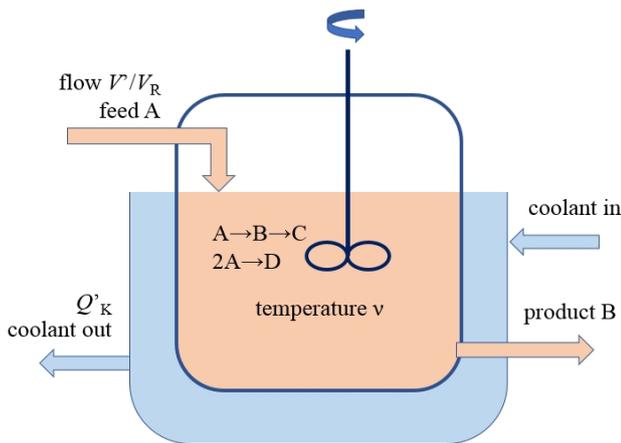


Fig. 2. Van der Vusse CSTR with a cooling jacket.

characteristics such as change of sign of gain in steady state behaviour and changes in zero dynamics.

The main reaction scheme is the van der Vusse reaction $A \rightarrow B \rightarrow C$, $2A \rightarrow D$. The control inputs are the normalized flow to the reactor \dot{V}/V_R and heat \dot{Q}_K withdrawn from the coolant by an external heat exchanger.

System dynamics are given by nonlinear ordinary differential equations (Chen et al. (1995); Ikonen et al. (2016)) obtained from component balances for the substances A and B, concentrations c_A and c_B [mol/l]

$$\begin{aligned} \dot{c}_A &= \frac{\dot{V}}{V_R} (c_{A0} - c_A) - k_1 c_A - k_3 c_A^2 \\ \dot{c}_B &= \frac{\dot{V}}{V_R} c_B + k_1 c_A - k_2 c_B \end{aligned} \quad (10)$$

and energy balances for the reactor and cooling jacket in temperatures ν and ν_K [$^{\circ}\text{C}$]

$$\begin{aligned} \dot{\nu} &= \frac{\dot{V}}{V_R} (v_0 - \nu) - \frac{1}{\rho C_p} \times \\ &\quad \left(k_1 c_A \Delta H_{R_{AB}} + k_2 c_B \Delta H_{R_{BC}} + k_3 c_A^2 \Delta H_{R_{AD}} \right) \\ &\quad + \frac{k_w A_R}{\rho C_p V_R} (\nu_K - \nu) \\ \dot{\nu}_K &= \frac{1}{m_K C_{PK}} \left(\dot{Q}_K + k_w A_R (\nu - \nu_K) \right) \end{aligned} \quad (11)$$

Many of the system physical parameters, such as reaction velocities k_i , reaction enthalpies ΔH_R , heat capacity C_p , heat transfer coefficient k_w , coolant heat capacity C_{PK} and density ρ are known only within bounds. Fixed parameters include reactor mass m_K , volume V_R and cooling surface A_R . A complete description of the equations and parameters can be found from Chen et al. (1995). The product concentration c_B and reactor temperature ν are measured as outputs. The benchmark uses a sampling interval of 20 seconds.

All plant measurement data was generated using the basic parameter values (see Chen et al. (1995), Table 1). The simulation inputs consisted of random ramps with random set point changes at random times at both control inputs. The start of a ramp was a random event taking place

with a probability of 20 seconds/5 hours. The length of the ramp ranged from 20 seconds to 20 minutes; the amplitude of the ramps ranged between minimum and maximum values, $U(3,35)$ for \dot{V}/V_R and $U(-9000,0)$ for \dot{Q}_K . The feed temperature was considered to be a constant at $\nu_0 = 104.9$ [$^{\circ}\text{C}$]. Data sets of length two days, two months and one year were considered for parameter estimation. A separate data set of 1 year was generated for testing purposes. In most of the experiments, the data was resampled to a sampling time of 2 min, with no averaging. Both output measurements were corrupted by additive zero mean normal noise with standard deviation corresponding to approximately 1% of the range of the signal, $\sigma_{c_B} = 0.01$ and $\sigma_{\nu} = 1$, respectively. In all, the input data sequence can be considered as rich for an industrial plant, but does not represent a possible scenario.

The physical model PM was taken to be the worst case scenario 1 (see Chen et al. (1995), Table 2), significantly different from the basic values. In some of the experiments the scenario 2 was considered (see Chen et al. (1995), Table 2). These two scenarios were considered as the physical extreme cases in Chen et al. (1995), representing the worst-case deviations from the basic model. The steady-state solution for the worst case 1 was considered as the PM in one setup. A FIR filter length of 20 min for c_B and 2 hours for ν samples was used for all filters, leading to FIR orders $n = (10, 60)$ for c_B and ν , respectively. In some of the experiments the AIC optima were used, $n = (1, 38)$ for the 2 month data set and 2 min sampling time. With steady state PM, slightly longer windows $n = (30, 90)$ were used. The setup with PM equal to 2 month data from the plant sampled every 2 min, PM as the model for the worst case scenario 1, and filter orders $n = (10, 60)$ will be referred to as the ‘nominal case’.

The number of PMFIR parameters to estimate is $1 + np + (1 + n)$, for each output. In the nominal set up, this results in 32 parameters for c_B and 182 for ν . All estimations were conducted by first normalizing the data to be zero mean with unit variance and using then a regularized LS with $r = 0.1$ and weight equal to the delay in input (in samples). This choice of algorithm, hinted in Nikolaou and Vuthandam (1998), together with the resampling from 20 sec to 2 min, resulted in a relatively smooth step responses in the estimates and very feasible computing times. Robust estimation of FIR/FSR models in the context of industrial processes is a rich topic in itself but not the content of this paper.

5.1 Nominal case

The nominal data consisted of 43200 sampled (\mathbf{u}, \mathbf{y}) pairs of inputs \dot{V}/V_R and \dot{Q}_K and outputs c_B and ν , simulated with parameters from the nominal scenario. The plant model was taken to be the worst case scenario 1, and used for simulating the same input sequence, resulting in the PM prediction. The CFIR and EFIR parameters were then estimated using the data set. The Bode diagram (gain and phase) for the CFIR filters is shown in Fig 3. The correction of PM for c_B involves a very slight lead, while for ν the filter is a low pass. The evaluation of the impact of CFIR tuning is immediate and precise via a Bode diagram. Similar plots can be drawn for EFIR’s.

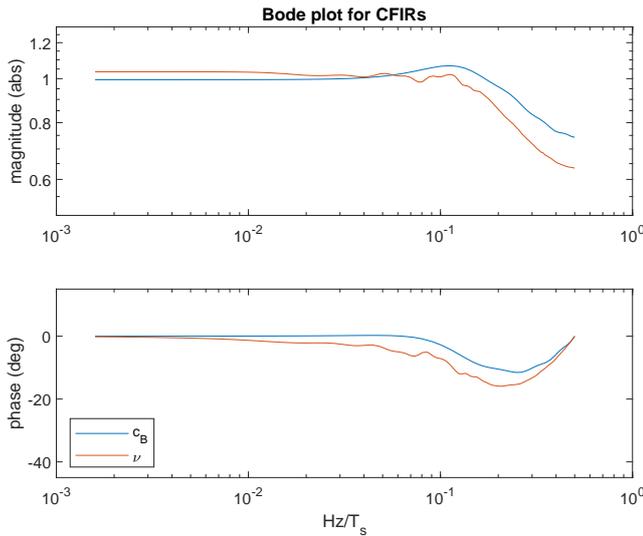


Fig. 3. Bode diagrams for estimated CFIR filters.

Figures 4–5 illustrate an alternative view, by showing the unit step responses for the filters. For CFIR (top plots in Figs. 4–5), a unit step would imply that the tuning has not modified the output of the physical model. It is clear that some dynamics are introduced, as already visible from the Bode (Fig. 3). Especially the CFIR response for ν (Fig. 5) resembles that of a low pass filter. One can analyze the static behaviour of the estimated filters by approximate affine mappings, by extracting b_0 's (shift) and summing the FIR coefficients. In the nominal case CFIR's, $y = 0.032 + 0.99y_p$ was obtained for c_B and $y = -1.191 + 1.035y_p$ for ν . Hence, the static gains were close to 1, as was already visible from the Bode (Fig. 3). However, these corrections were significant, as these shifts account for over 0.03 mol/l in product concentration and one degree Celsius in reactor temperature.

Similar dynamic and static summary info can be obtained for EFIR's. For EFIR's, a zero step indicates no impact. Looking at Figs. 4–5 (middle and bottom plots), the absolute values of the estimated step responses remain very close to zero at all times. The response in ν contains some variation difficult to explain (see Fig. 5 bottom plot), potentially indicating problems in reliably estimating such a large number of parameters from data. However, the scales are of the order 10^{-4} and 10^{-6} . Again, the analysis of tuning outcomes is straightforward, albeit with EFIR the impact of the scales of the input signals needs to be taken into account when assessing the unit steps.

Figure 6 illustrates the predictions for ν in a simulation over the test set. The measured data is noisy (standard deviation 1 °C). The shift error in PM is clearly visible, and it is corrected by the PMFIR. Corrections in the dynamic response are not visible by eye in the one-day simulation interval, these are better quantified from the Bode, Fig. 3.

5.2 Further analysis

A series of tests was conducted to better illustrate and quantify the impact of various components in the PMFIR approach and to compare it with alternative approaches. Starting from the nominal case, changes were made in

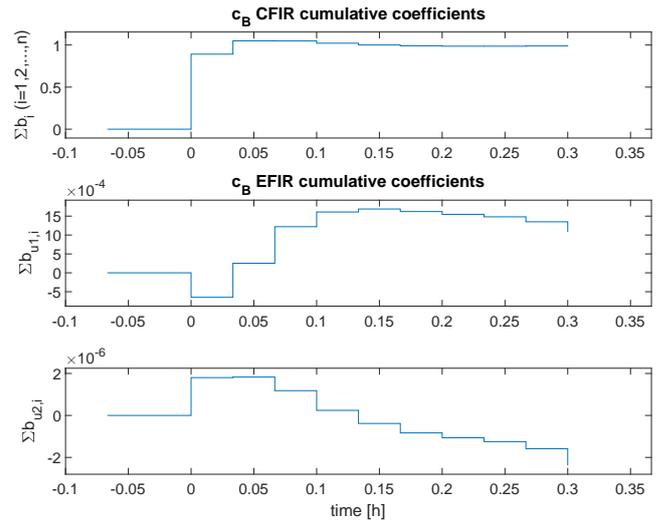


Fig. 4. Estimated PMFIR coefficients for output c_B .

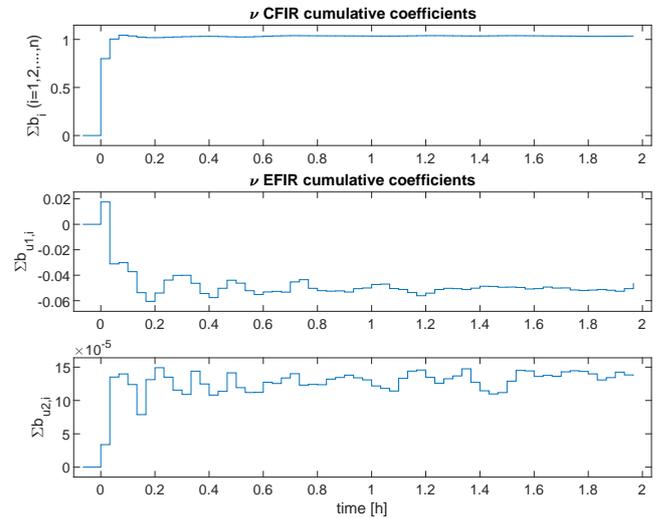


Fig. 5. Estimated PMFIR coefficients for output ν .

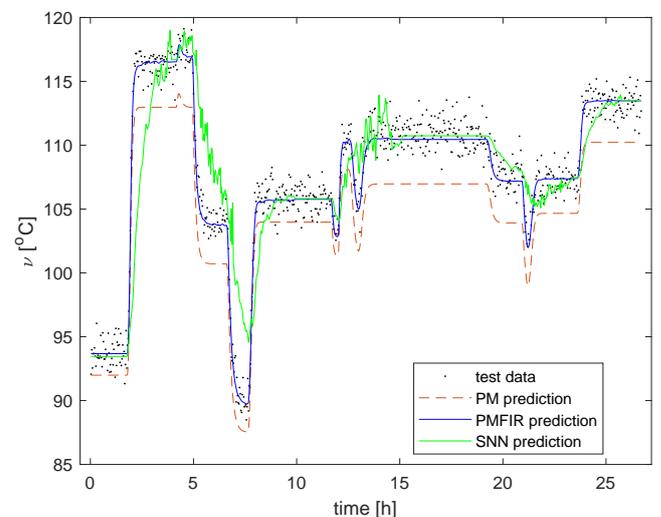


Fig. 6. Predictions of ν on a short interval of the test set.

the amount of data samples, sampling rate, PM and filter orders. The approach was also compared with a standard sigmoid neural network (SNN) in a similar FIR time series setup. As a main metric, RMSE (root mean squared errors) were computed both on training and test data.

The test series consisted of 8 variations to the nominal setup, and three SNN topologies:

- nominal case: parameter estimation was conducted based on a 2 month data set with 2 min sampling interval. Physical model corresponds to scenario 1 (PM1).
- nominal case, but using a model corresponding to scenario 2 as a physical model (PM2)
- nominal case, but using a short data set of 2 days, only.
- nominal case, but using a short data set and physical model of scenario 2.
- nominal case, but using a long data set of 1 year data.
- nominal case, but using a long data set and physical model of scenario 2.
- nominal case, but using the AIC-optimal orders for the FIR-filters $n = (1, 38)$
- nominal case, but using a sampling time of 20 sec and AIC optimal order for FIR-filters $n = (1, 59)$
- nominal case, but using an approximate static physical model for scenario 1 (PM3) and a higher order of filters $n = (30, 90)$
- sigmoid neural network (SNN) with $\{3, 8, 15\}$ nodes in the hidden layer, trained using 2 day data
- SNN with $\{3, 8, 15\}$ nodes in the hidden layer, trained using 2 month data
- SNN with $\{3, 8, 15\}$ nodes in the hidden layer, trained using 1 year data

For each of the above PMFIR setups, four RMSE were computed:

- RMSE between measured data and the physical model, $PM \in \{PM1, PM2, PM3\}$
- RMSE between measured data and a static affine model from the PM output: $\hat{y} = ay_p + b$ with coefficients a and b .
- RMSE between measured data and the CFIR prediction
- RMSE between measured data and the PMFIR prediction, i.e., CFIR + EFIR.

For SNN setups, only the RMSE between measured data and the SNN prediction was computed.

The test series allows to assess the impact of the amount of data, sampling, model order and modeling type. Looking at Figs. 7–8 it can be seen that on a short data set the CFIR and PMFIR can closely mimic the training set, on ν the RMSE on training data is even smaller than the noise component $N(0,1)$. The RMSE increases on test set, however, indicating mild overlearning. This is not observed with 2 month or 1 year data sets, the RMSE of which are in many cases not distinguishable. There is a clear difference between the two extreme scenarios PM1 and PM2, which is visible in particular with c_B . A better PM results also to a better PMFIR RMSE. The RMSE are insensitive to differences in nominal case order and AIC-optimal orders. This is significant especially with c_B where

the AIC-optimized order was one. Hence, the FIR-window was not beneficial from the point of view of improving the prediction RMSE. The result is confirmed by the small differences between the affine static modeling and CFIR modeling with both c_B and ν . This observation indicates that the dynamics of the worst case scenarios (used as plant models) are in fact not so far from the true ones.

A completely different result is depicted by observing the RMSE for the case of static PM (PM3). A steady state PM is a special case of process models. In PM3, the physical model of the van der Vusse CSTR plant was replaced by its approximate steady state description, i.e.

$$\begin{aligned} \mathbf{x}(k+1) &= f_{PM-ss}(\mathbf{u}(k)) \\ \mathbf{y}_{PM}(k) &= g_{PM}(\mathbf{x}(k)) \end{aligned} \quad (12)$$

where f_{PM-ss} is the steady state solution of the van der Vusse CSTR model. In series with CFIR, it leads to Hammerstein structures, commonly used in the industry. Due to lack of dynamics in the PM, larger filter orders were chosen: $n' = (30, 90)$ corresponding to 1 and 3 hour windows (the AIC optimized orders were $n^* = (9, 705)$). Observing Figs. 7–8, while static affine mapping of the PM does reduce the RMSE from approximately 0.85 to 0.75 for c_B (4.5 to 3.5 for ν), the addition of FIR dynamics to the static mapping reduces it further to 0.5 (below 2). The overall RMSE's on the PMFIR based on steady state PM still remain rather unsatisfactory, however, as the van der Vusse dynamics are strongly nonlinear.

The static affine mapping can be considered as an alternative to PMFIR, with the advantages of being particularly robust to estimate and simple to implement. Clearly, it is very useful if the plant dynamics are well modelled by the physical model, and only a static adjustment is needed.

An alternative to nonlinear dynamic modeling is a standard feedforward sigmoid neural network (Ikonen and Najim (2002), Sec 4.2.2; Hastie et al. (2017), Sec. 11.3; Shoukens and Ljung (2019)). It is a well known representative of the long list of approaches for data-driven modeling. The nonlinear dynamic modeling problem is a typical area of application for SNN, used in various NARX, NOE, NFIR, etc., time-series structures (N refers to nonlinear). For comparison, FIR SNN time-series model structures were constructed. The inputs consisted of the delayed $n = (10, 60)$ inputs (normalized flows and heat withdrawn from the coolant), sampled at 2 min. The one-hidden layer (NFIR) structure consisted of h hidden nodes, $h = \{3, 8, 15\}$ with a bias term, and a linear output node with bias term. The SNN were trained using the default setup of Matlab Deep Learning Toolbox *feedforwardnet* and *train* functions, using Levenberg–Marquardt for parameter estimation and reservation of 15 % of training data for early stopping based on performance on validation set. The best-performing SNN structure (in terms of performance on test set) is reported for each data set (see legends in Figs. 7–8). Also alternative FIR SNN structures were experimented, where inputs consisted of smaller subsets of the full window, but the conclusions remained the same.

In general, the data-driven modelling easily suffers from lack of data when system dimensions increase. Looking at Figs. 7–8, the training on short data set seems to provide promising results, but the trained SNN perform poorly

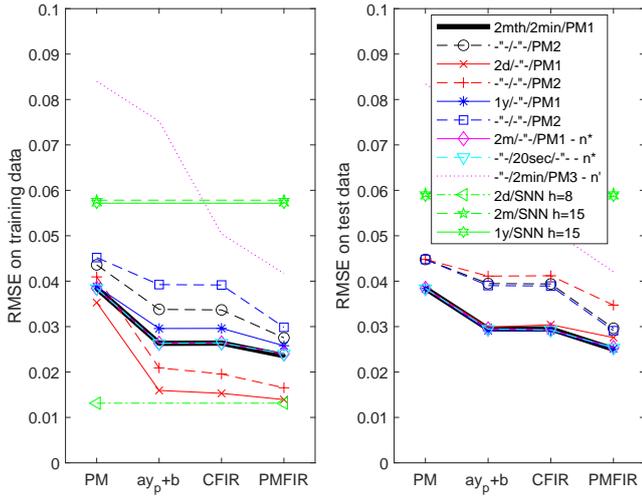


Fig. 7. RMSE between measured c_B and predictions by various approaches.

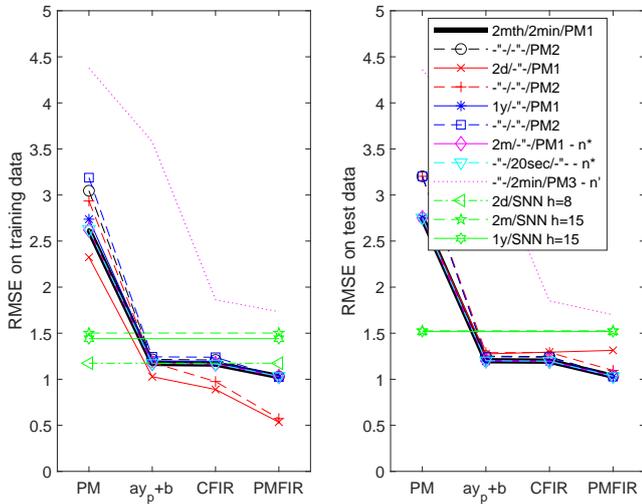


Fig. 8. RMSE between measured ν and predictions by various approaches.

on test data. For two day data SNN case, the RMSE on test set is so high it goes outside the window (RMSE 0.26 and 7.7 for c_B and ν , respectively). Figure 6 shows also a prediction given by the SNN. The figure seems to indicate that the static behaviour has been captured somewhat but that the combination of a large dimension of the delayed inputs and sparsity of the data did not lead to meaningful learning outcomes of system dynamics. These results are in line with those obtained using the finite state FIR (FFIR) (Ikonen and Selek (2020)). This could be remedied to some extent by replacing the (nonlinear) FIR by an alternative time series approach. For example, the NOE is known to perform well (Ikonen and Najim (2002); Castro et al. (2010)) but is complicated by the need to compute the impact of model dynamics to gradients, and the requirements it poses on data and system stability.

5.3 State estimation

In the plant data used in the above simulations, the feed temperature v_0 was constant at 104.9 °C. As there is no information (measured data) of the impact of feed

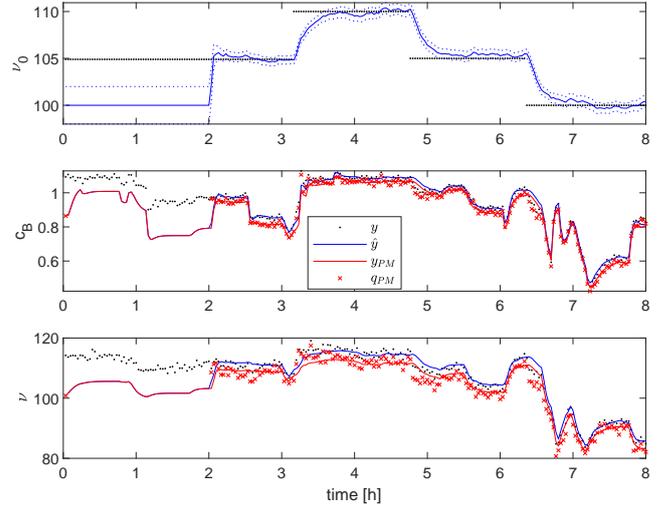


Fig. 9. Estimation of feed temperature ν_0 using UKF.

temperature to system outputs, any data-driven approach based on the above data sets would fail in estimating the effect of a change in this temperature. Since this variable is available in the plant physical model (f_{PM}, g_{PM}), it is straightforward to extrapolate the impact of ν_0 in a PM based approach. The extrapolations also carry to the calibrated model. The same applies to all variables/constants in the PM model. This simple case example again motivates taking full advantage of available physical models, as supported by the PMFIR approach.

An unscented Kalman filter (UKF) was constructed to illustrate the state estimation in the van der Vusse CSTR example. The unmeasured feed temperature was changed between $\{99.9, 104.9, 109.9\}$ °C. An augmented state space was defined by $\mathbf{x} = [c_A, c_B, \nu, \nu_K, \nu_0]'$ and an UKF set to estimate all the states using the PMFIR estimated model for the nominal case. The simulations were started with an incorrect initial mean ($[2, 1, 100, 100, 100]'$ [mol/l and °C]) with covariance ($\text{diag}[.1, .1, 1, 1, 1]^2$). The covariances of the zero-mean noise components were picked as $Q = \frac{1}{100}\text{diag}[0.01, 0.01, 2, 2, 1]^2$ for system state propagation, measurement noise characteristics were assumed to be correctly known, $R = \text{diag}[0.01, 1]^2$.

The estimated feed temperatures are shown in Fig. 9. Three stepwise disturbances impacted to ν_0 during the simulation (see top plot). The estimation is started at 2 hours from initialization (a sufficient input history is required by PMFIR). All states were correctly estimated by UKF, Fig. 9 top plot shows the estimated feed temperature ν_0 . The innovation signal ($y_{PM} - q_{PM}$) for both outputs is illustrated in the other plots, together with the PMFIR prediction and measured data.

6. DISCUSSION AND CONCLUSIONS

The application of physical models is widespread in many fields of engineering. These models need to be tuned due to various reasons, such as correction of simplifications/idealizations made during modeling or adaptation to systematic errors in plant measurements. Calibration of physical models is constrained by several engineering requirements, such as emphasizing a clear distinction between the original physical model and its tuning, auto-

maticity from data, and simplicity in implementation. In process control and state estimation, system dynamics are of particular interest.

The proposed approach can be seen as a machine learning tool fusing process models with measured data (Qin and Chiang (2019); Bismukhametov and Jäschke (2020)). The requirements for such methods in industrial process engineering were widely discussed in the paper, emphasizing the aspect of physical model tuning and that a model structure founded on a physical model can be expected to extrapolate much better to operation regions not visited in real life, i.e., for which no data is available. These are typically of interest when optimizing for improved plant performance or describing faulty or abnormal situations.

This work proposed a method for calibrating an existing physical simulation model. In the suggested PMFIR tool, the main characteristics of the mapping originate from the physical model. The CFIR provides a linear dynamic adjustment for the physical model outputs, the EFIR extends the tuning by a linear dynamic mapping from past inputs, thereby enabling identification of linear components not modelled by the physical model. The approach can be automatized with high robustness, and recursive implementations of the algorithms are readily available.

A typical problem with physical models of industrial processes is that they corrupt over time. Consequently, the models eventually become useless unless updated by a specialized process modeling expert. A priori, identification / machine learning techniques can be readily used for adaptive data-driven modeling. However, it is rare that re-identification of such models could be constructed automatically, but a data scientist is required. If the plant varies (slowly) with time, the linear PMFIR corrections (given some forgetting scheme is used) can make global corrections to the response. It can be expected that such procedure could be automated in a robust fashion.

The suggested PMFIR approach was illustrated and experimented with using a well known process control design benchmark of a highly nonlinear CSTR with a cooling jacket. The incorrect (worst-case) models were successfully calibrated using nominal plant data and even a steady-state PM could be successfully applied. The approach was compared with alternative data-driven methods, including static affine correction at the physical model output, and purely data-driven modeling of the dynamic process. The proposed approach provided improved performance and was shown to be much less sensitive to the amount of data than corresponding data-driven approaches. Finally, the application of the calibrated model in state estimation using UKF was successfully demonstrated.

On-going work focuses at robust estimation of linear dynamics and approaches for merging predictions from several plant models. In applied work, the focus is in gaining experiences on the potential in applications in the heavy process industry.

REFERENCES

- T. Bismukhametov and J. Jäschke. Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models. *Computers and Chemical Engineering*, vol. 138, pp. 1–27, 2020.
- L. Castro, O. Agamennoni and M. Alvarez. From linear to nonlinear identification: one step at a time. *Mathematical and Computer Modelling*, vol. 51, pp. 473–486.
- H. Chen, A. Gremling and F. Allgöwer. Nonlinear predictive control of a benchmark CSTR. *European Control Conference 1995*, pp. 3247–3252.
- C. Cutler and B.L. Ramaker. Dynamic matrix control - A computer control algorithm. *National meeting of AIChE*, April 1979.
- R. Das. *Mathematical Cell Biology Graduate Summer Course (Week 4, Lecture 3)*. University of British Columbia, Vancouver, May 1-31, 2012.
- S. Engel and K.-U. Klatt. Nonlinear control of a non-minimum-phase CSTR. *Proceedings of the American Control Conference*, San Francisco, pp. 2941–2945, 1993.
- C. Garcia and A. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, vol. 46, pp. 73–87, 1986.
- W. Garcia-Gabin, J. Normey-Rico and E. Camacho. Sliding mode predictive control of a delayed CSTR 6th IFAC Workshop on time-delay systems, *IFAC Proceedings*, vol. 39 Issue 10, pp. 246–251, 2006.
- P. Gatti. *Probability theory and mathematical statistics for engineers*. London: Spon Press, 2005.
- E. Gatzke and F. Doyle III. Multiple model approach for CSTR control. *newblock IFAC Proceedings*, vol 32, pp. 6950–6955, 1999.
- N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings-F*, vol. 140, no 2, pp. 107–113, 1993.
- T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning - Data mining, inference and prediction*. 2nd ed, Springer, 2017.
- E. Ikonen, and K. Najim. Nonlinear process modeling based on a Wiener approach. *Proceedings of the Institute of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 215, no 1, 15–27, 2001.
- E. Ikonen, and K. Najim. *Advanced Process Identification and Control*. New York: Marcel Dekker, 2002.
- E. Ikonen, I. Selek and K. Najim. Process control using finite Markov chains with iterative clustering. *Computers and Chemical Engineering*, vol. 93, pp. 293–308, 2016.
- E. Ikonen and I. Selek. Calibration of physical models with process data using FIR filtering. *Australian and New Zealand Control Conference*, 2020.
- S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. *American Control Conference*, 1620–1632, 1995.
- C. Kravaris, M. Nuenec, R. Berber and C. Brosilov. Nonlinear model-based control of nonminimum phase processes. In: R. Berber and C. Kravaris (eds.) *Nonlinear Model Based Process Control*, 115–142, Kluwer Academic Publishers, 1998.
- M. Lawrynczuk. Nonlinear predictive control of dynamic systems represented by Wiener–Hammerstein models. *Nonlinear Dynamics*, vol. 86, pp. 1193–1214, 2016.
- L. Ljung. *System Identification*. Technical Report from Automatic Control LiTH-ISY-R-2809, 1997.
- P. Marusak. Numerically efficient fuzzy MPC algorithm with advanced generation of prediction – Application to a chemical reactor. *Algorithms*, vol. 13, issue 6, 2020.

- M. Nikolaou and P. Vuthandam. FIR model identification: Parsimony through Kernel compression with Wavelets. *AICHE Journal*, vol. 44, pp. 141–150, 1998.
- H. Perez, B. Ogunnaike and S. Devasia. Output tracking between operating points for nonlinear processes: Van der Vusse example. *IEEE Transaction on Control System Technology*, vol 10, pp. 611–617, 2002.
- S. J. Qin and L. H. Chiang. Advances and opportunities in machine learning for process data analytics. *Computers and Chemical Engineering*, vol. 126, pp. 465–473.
- J. Richalet, A. Rault, J. Testud and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, vol. 14, pp. 413–428.
- J. Shoukens and L. Ljung. Nonlinear system identification: a user oriented road map. *Control Systems Magazine*, vol. 39, pp. 28–99, 2019.
- D. Simon. *Optimal State Estimation - Kalman, Hinf and nonlinear approaches*. John Wiley, 2006.
- J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Dylon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, vol. 31, pp. 1691–1724, 1995.
- S. Skogestad. *Chemical and Energy Process Engineering*. CRC Press, 2009.
- B. J. Spivey, J. Hendegren, and T. Edgar. Constrained nonlinear estimation for industrial process fouling. *Industrial and Engineering Chemistry Research*, vol 49, pp. 7824–7831, 2010.